An Abstraction Method Using a Semantic Engine Based on Language Information Structure

Hirohito INAGAKI and Tohru NAKAGAWA

NTT Human Interface Laboratories
1-2356 Take Yokosuka-Shi Kanagawa 238-03 Japan
E-mail inagaki@ntthli.ntt.jp

Abstract

This paper describes the framework for a new abstraction method that utilizes event-units written in sentences. Event-units are expressed in Language Information Structure (LIS) form and the projection of LIS from a sentence is performed by a semantic engine. ABEX (ABstraction EXtraction system) utilizes the LIS output of the semantic engine. ABEX can extract events from sentences and classify them. Since ABEX and the LIS form use only limited knowledge , the system need not construct or maintain a large amount of knowledge.

1 Introduction

Automatic abstraction, a major natural language processing application, is difficult to achieve. Luhn[5] developed a very simple extraction method that searches for the keywords in each sentence. This type of abstraction is easy to accomplish, but its quality is poor. Other abstraction methods[6, 3] utilize natural language understanding (NLU). However NLU is still in the development stage. For achieving good practical performance, it is necessary to treat the information expressed in a document uniformly so that it can be analyzed with only a small fixed amount of knowledge.

We propose the LIS form which allows information about events to be uniformly treated. Furthermore, the semantic engine only uses abstract words, this reduces the size of the knowledge. So the semantic engine projects a sentence to a LIS form within a limited amount of knowledge.

In abstraction, classification is the first step. Classification is performed using uniform approach, the LIS form. LIS event representation allow us to select and classify sentences.

2 A semantic engine

2.1 What is the Language Information Structure(LIS) form?

The LIS form expresses the information structure that permits communication between individuals. If two individuals communicate about one that happened (will happen) in the real world, the core information is the event. Sometimes a speaker will attache an attitude to the event. So information about real world is expressed by the event and the attitude of the speaker.

2.2 LIS form

In the LIS form, there are two types of feature-structure, word feature-structure and event feature-structure. Almost all slots of the word feature-structure are filled with appropriate values and few slots of the event feature-structure are empty. The semantic engine tries to fill all slots of the feature-structure.

2.2.1 Event feature-structure

Event has one feature-structure, the role feature. The sentence contains one or more events and the event feature-structure indicates the role of words or phrases in the event. The role feature is either essential or extensional. Seven essential roles have been created: as AGENT, OBJECT, ACTION, LOCATION, TIME, FROM, and TO. These roles are defined not for verbs but for events. This is quite different from Fillmore's cases [2]. Therefore, the action in the event is represented by the ACTION slot, which can be filled by verbs, nouns, gerunds, and so on. It is not necessary to fill the ACTION slot by a verb.

For example, the phrase " a land purchase agreement" is dealt with as one event in the LIS, and the ACTION slot-value is "agreement". Other slots, such as AGENT, OBJECT, LOCATION, TIME, FROM and TO slots are almost the same as in Fillmore's cases of 'agent', 'object', 'location', 'time', 'source', and 'goal (or experiencer)'. It is important that our role model deals with the roles of words (or phrases) in an event, not word meaning.

Using just seven essential roles, it is difficult to assign a role to a word (or a phrase). To overcome this problem, we introduce extensional roles which allow to be modified by the addition of "/constraint".

2.2.2 Word feature-structure

Word has six features. These are semantic feature (DDF) slot, numerical-value slot, date slot, constraint slot, modality slot, and word string slot.

Using the semantic feature, the event feature-structure will be determined during semantic interpretation process. Six classes of semantic features are defined, such as INDIVIDUAL, ELEMENT, THING, ACTION, LOCATION, and TIME These classes are instantiated to the Domain Dependent semantic Features (DDF) when the domain is decided.

The constraint feature restricts the feature-structure of brother words or phrases. Furthermore, the constraint feature determines the relations between word feature-structure and event feature-structure. In Japanese language, a word which have a ACTION DDF usually has the constraint feature that determines the slots of event feature-structure.

The numerical-value slot expresses numerical value of a word ; 0, 1 , 2, 一 (one) , 百 (hundred) , 千 (thousand) , and so on. The calculation of counting up and down is necessary, so all figures are separated. The numerical-value feature will be expressed as follows.(Our notation of a feature-structure is [*feature-name* = *feature-value*].)

$$\left[\textit{numerical-value} = \begin{bmatrix} \textit{first-digit} & = & \cdots \\ \textit{second-digit} & = & \cdots \\ \cdots & = & \cdots \end{bmatrix} \right]$$

The date slot expresses event occurrence time and is expressed by the Christian era. In the Christian era, days are counted by numbers, so that date slots are calculated using the numerical-value feature. The date slot has a minute slot, a second slot, a hour slot, a day slot, a month slot, and a year slot. Each slot is expressed in numerical value.

The modality slot is classified into three categories; tense, aspect, and mood. Since the tense and aspect are linguistically fixed, we use an ordinary categorization. However, mood is needed to be categorized differently, because the information unit used this system is an event. So we categorized mood as a combination of Bratman's Belief-Desire-Intention model[1] and modal logic. That is the state of event is expressed by modal logic (necessary operator, possible operator, and negation sign) and the attitude of speakers can be classified into belief, desire, and intention. For example, a sentence *I think it is possible to construct a plant there* will be expressed as $Belief(Possible E)$, where E means an event;*construct a plant there*, that is, the individual believes that E is possible.

Furthermore, it is necessary to consider a situation in which information is transferred. In the newspaper, it is created by journalists who get information from other services (person or company information bureau). In this situation, the event and the attitude of the information possessor (IP) is transported to a speaker (SP);journalist. The journalist then reexpresses the information to reflect his attitude. If the modality of IP and SP are expressed as $Modality_{SP}()$, $Modality_{IP}()$, respectively, information in newspaper is expressed as,

$Modality_{SP}(Modality_{IP}(EVENT)))$.

If the target document is newspaper, the LIS form includes the modality of speaker ($Modality_{SP}$) and the modality of information possessor ($Modality_{IP}$).

2.3 Projection Mechanism

Parsing is done using Morphological analysis and Dependency analysis[4] and yields a syntactic tree for a sentence.

After the parsing, we search a feature-structure dictionary to extract feature-structures of all words related to the domain. To perform semantic analysis with limited knowledge, word feature-structures are prepared only for abstract words. The registration of proper nouns are left to the user . The semantic engine infers the semantic meaning of words or phrases from the system default and user registrations held in the dictionary. This means the semantic engine do not need all knowledge of words for semantic interpretation. Thus only a small amount of words need to be maintained.

After attaching the appropriate word feature-structure to all important words, semantic interpretation can proceed. From the type of propagation for the feature-structure in a parsing tree,there are two types of features. One is the synthesized type whose value is calculated from sons to father relationship of the parsing tree. The other is the inherited type that are calculated from father or brothers.

Word string, DDF, numerical-value, date, and modality features are synthesized type and other features, such as constraint and role are inherited type. The propagation of a feature-structure is accomplished by unification calculus, but the grammar is different.

For DDF features, the grammar is as follows.

$$E.DDF ::= indefinite$$

$$N.DDF ::= N_1.DDF \oplus N_2.DDF \oplus \cdots N_n.DDF$$

$$N.DDF ::= individual(company) | element(company) | money | man | product | action(company) | location | time$$

Note: Uncapitalized words mean terminate and capitalized words mean nonterminate. E means EVENT node structure and N means other node structure and N.DDF means DDF feature-value in node 'N'. Symbol 'n' is the number of nodes. Operator ⊕ means unification operator.

For the constraint feature,

$$E.constraint ::= \sum_{i=1,\cdots,n} (N_i.DDF \oplus E)$$

$$N.constraint ::= \sum_{j=1,\cdots,n} (N_j.DDF \oplus N)$$

$N.constraint ::=$ *feature-structure of brother nodes*

The date and numerical-value features are rather complicated because we have to deal with the semantic meaning of time.

The grammar of for the date feature is,

$$N.date ::= N_1.date \oplus N_2.date \oplus \cdots \oplus N_n.date$$

$$N.date ::= \begin{bmatrix} minute & = & \cdots \\ second & = & \cdots \\ hour & = & \cdots \\ day & = & \cdots \\ month & = & \cdots \\ year & = & \cdots \end{bmatrix}$$

The calculation of number and date features is done like a stack. The numerical-value feature has one stack and date feature has six stacks.

For example, for the number *1992*, all the numbers, 1, 2, and 9, are expressed as follows,

1 → [numerical-value = (eval (push-number-stack 1))]

2 → [numerical-value = (eval (push-number-stack 2))]

9 → [numerical-value = (eval (push-number-stack 9))]

Note: Symbol 'eval' means that next form will be evaluated by Common-lisp. Symbol 'push-stack' is the function that puts the argument value on the top of the stack.

The equation for the numerical-value of *1992* is,
1992.numerical-value =
{[numerical-value = (eval (push-number-stack 1))]
⊕[numerical-value =(eval (push-number-stack 9))]
⊕[numerical-value = (eval (push-number-stack 9))]
⊕[numerical-value =(eval (push-number-stack 2))]}

which, after evaluation, gives as the value of *1992* as follwing expressions,couting right to left, first digit being 2, second digit begin 9, and so on.

$$1992.numerical\text{-}value = \begin{bmatrix} first\text{-}digit & = & 2 \\ second\text{-}digit & = & 9 \\ third\text{-}digit & = & 9 \\ fourth\text{-}digit & = & 1 \end{bmatrix}$$

If we process the phrase, *1992 年 (year 1992)*, the equation becomes,

1992 年.date =

$$\left\{ 1992.numerical\text{-}value \oplus \begin{bmatrix} date & = & (eval\ (if\ SELF.numerical\text{-}value \\ & & (push\text{-}year\text{-}stack\ SELF.numerical\text{-}value))) \end{bmatrix} \right\}$$

Note: Nonterminate 'SELF' refers to the self feature-structure. Symbol 'push-year-stack' is the function that places the argument value to the year stack in the date feature.

Then and we get the time feature-structure as,

$$[\ 1992\text{年}.date = [\ year = 1992\ ]\ ]$$

The grammar for modality is quite simple,

$$E.modality ::= N_1.modality \oplus N_2.modality \oplus \cdots N_n.modality$$

$$N.modality ::= N_1.modality \oplus N_2.modality \oplus \cdots N_n.modality$$

$$N.modality ::= tense, aspect, and\ mood$$

2.4 An example of the projection process in the semantic engine

This passage comes from The Nikkan-kougyou shinbun(Daily Industrial Newspaper). The headline is "高田機工、和歌山に大型橋梁の新工場建設"(Takada Kiko Co., Ltd. is constructing a new plant to assemble large-scale steel bridges in Wakayama-ken.).

S1: "高田機工は十六日、橋梁の大型化に対応して和歌山県下津町に進出し、新工場を建設することで用地売買契約に調印したと発表した。" (*Takada Kiko announced a land purchase agreement in Shimotsu-machi, Wakayama-ken, where they will construct a new plant giving them additional space and capabilities to fabricate larger scale steel bridge structures.*)

S2: "建設用地は旧丸善石油精製工場跡地の一部を、丸善下津興産から土地十七万八百五十六平方メートルを約百億円で買収、来春から新工場建設に着工する。" (*The 170,856-sq.-meter construction site, previously occupied by part of a Maruzen Oil Co., Ltd. refinery, was purchased from Maruzen Shimotsu Kosan for an estimated ¥10 billion. Construction on the new plant facility is slated to begin this coming spring.*)

S3: "投資額は約二百二十億円。" (*Investment capital is about ¥22 billion.*)

S4: "平成五年四月操業の予定。" (*Operations will begin in April, 1993.*)

(The Nikkan-kougyou shinbun, November 17, 1990)

At first, the domain is decided. In this example, we use "企業関係"(company act) domain.

The system roughly separates events and extracts all events which related to the domain: company act. From the feature-structure dictionary, the sentence was quickly reviewed to determine whether there is a word which has an ACTION DDF or not. If there is no such word, the system then stops analyzing the sentence. In the example, the first sentence (S1) is made of two events. One is "建設 (construction)" and the other is "調印 (agreement)". The two events are connected by "することで (surukoto-de)". So sentence (S1) is separated to two events. Sentence S3 does not have an ACTION DDF, so further analysis is not conducted.

Therefore, we can obtain five events from five sentences; S1 to S5.

Event1inS1: "高田機工は十六日、橋梁の大型化に対応して和歌山県下津町に進出し、" (*Takada-kiko Co., Ltd. will construct a new plant giving them additional space and capabilities to fabricate larger scale steel bridge structures.*)

Event2inS1: "新工場を建設することで用地売買契約に調印したと発表した。" (*Takada Kiko announced a land purchase agreement*)

Event3inS2: "建設用地は旧丸善石油精製工場跡地の一部を、丸善下津興産から土地十七万八百五十六平方メートルを約百億円で買収、" (*The 170,856-sq.-meter construction site, previously occupied by part of* *a Maruzen Oil Co., Ltd. refinery, was purchased from Maruzen Shimotsu Kosan for an estimated ¥10 billion.)*

Event4inS2: "来春から新工場建設に着工する。" (*Construction on the new plant facility is slated to begin this coming spring.*)

Event5inS4: "平成五年四月操業の予定。" (*Operations will begin in April, 1993.*)

After the event separation process concludes, semantic interpretation is commenced. The first stage is attaching a feature structure to each word.

Let's consider the Event4 in S2; "来春から新工場建設に着工する。" (*Construction on the new plant facility is slated to begin this coming spring.*). In this passage, there are three Bunsetsu, five independent words, three dependent words. We need only five feature-structures as shown below.

来春;this coming spring—

$$\mathbf{u1}\left[\begin{array}{lcl} string & = & \text{"来春 (raisyun)"}; this\ comming\ spring \\ DDF & = & time \\ date & = & (push\text{-}year\text{-}stack\ (1 + *article\text{-}year*)) \end{array}\right]$$

工場;plant—

$$\mathbf{u2}\left[\begin{array}{lcl} string & = & \text{"工場 (koujyou)"}; plant \\ DDF & = & element(company) \end{array}\right]$$

建設;construction—

$$\mathbf{u3}\left[\begin{array}{ll} string = & \text{"建設 (kensetsu)"}; construction \\ DDF = & action(company) \\ \text{con-straint} = & \left[\begin{array}{lcl} \_agent.DDF & = & individual(company) \\ \_action.DDF & = & action(company) \\ \_object.DDF & = & element(company) \\ \_time.DDF & = & time \\ \ldots & = & \ldots \end{array}\right] \end{array}\right]$$

着工;is slated to ...—

$$\mathbf{u4}\left[\begin{array}{lcl} string & = & \text{"着工 (chakkou)"}; is\ slated\ to... \\ modality & = & [aspect = just\_before] \end{array}\right]$$

する;will—

$$\mathbf{u5}\left[\begin{array}{lcl} string & = & \text{"する (suru)"}; will \\ modality & = & [tense = future] \end{array}\right]$$

Note: Symbol '_agent.DDF' means that if the DDF feature value is unified to the one node, then variable '_agent' is bounded to that node's feature-structure. Variable *article-year* is bounded to the date of year when the article is published.

The example is parsed as shown in figure 1.

Once the parsing is finished, the semantic interpretation process begins. Node *n1* will have the feature-structure that is the result of calculation between the feature-structure of "来春 (raisyun)" and "から (kara)", but the word "から (kara)" has no feature-structure so the feature-structure of "来春 (raisyun)" *u1*, is propagated to node *n1*. The feature-structures of all nodes are calculated same way.

For the constraint feature, unification was done to all brothers. For example, $[\_agent.DDF =$

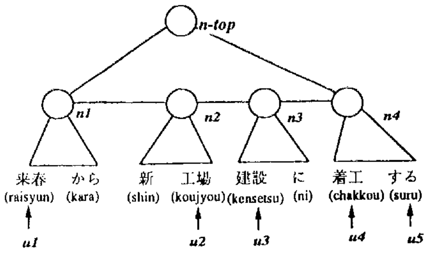

Figure 1: First stage of syntactic tree

$individual(company)]$ means that one brother node is needed which have the DDF value of *agent(company)*. If there is a node which satisfies the constraint, then the variable *_agent* is bounded to that node feature-structure. If there is no node which satisfies the constraint, then variable *_agent* is unbounded.

Try to think about the constraint feature in "建設 (kensetsu)". There is no node that has *agent(company)* in DDF, but there are nodes which satisfy the constraint, such as the *_action*, *_object*, and *_time* which are bounded to nodes *n3*, *n2*, *n1*, respectively.

Finally we get the event feature-structure of top node *n-top*, shown in figure 2.

3 An abstraction using the LIS form

3.1 The basic method of the abstraction

In the abstraction, we utilize classification of the LIS output. First, a sentence is put into the LIS form by the semantic engine.

The LIS output is used to commence the abstraction procedure. To extract information from sentence, we think classification is the best way. The semantic engine analyzes sentence in fixed domain. after the semantic analysis, Sentences are classified whether an event or not, and the system extracts the events which are related to the domain.

Finally, ABEX provides a abstraction. One abstraction proposed here is the classification of event occurrence time and similarity of event. This classification reveals the relationships of each event. Individual event occurrence times will be determined from value of the time feature and the similarity of events is calculated by comparing event feature-structure slots.

The other method is classification by the modality of information. From the view point of $Modality_{IP}$, we can classify an event according to the modality of information possessor (IP). If there is no modality in the event, we classify it as 'fact'. Others are classified using modality feature. This classification of the event's modality reveals the attitude of the information possessor.

$$
\left[\begin{array}{lcl}
\_action & =_{u3} & \left[\begin{array}{lcl} string & = & \text{“建設 (kensetsu) に (ni)”} \\ DDF & = & action(company) \\ \text{constraint} & = & \left[\begin{array}{lcl} \_action.DDF & = & action(company) \\ \_object.DDF & = & element(company) \\ \_time.DDF & = & time \end{array}\right] \end{array}\right] \\
\_object & =_{u2} & \left[\begin{array}{lcl} string & = & \text{“新 (shin) 工場 (koujyou)”} \\ DDF & = & element(company) \end{array}\right] \\
\_time & =_{u1} & \left[\begin{array}{lcl} string & = & \text{“来春 (raisyun) から (kara)”} \\ DDF & = & time \\ date & = & \left[\begin{array}{lcl} month & = & 4 \\ year & = & 1991 \end{array}\right] \end{array}\right] \\
\_ & =_{u4 \sqcup u5} & \left[\begin{array}{lcl} string & = & \text{“着工 (chakkou) する (suru)”} \\ modality & = & \left[\begin{array}{lcl} aspect & = & just\_before \\ tense & = & future \end{array}\right] \end{array}\right]
\end{array}\right]
$$

Figure 2: An example of an event node feature-structure

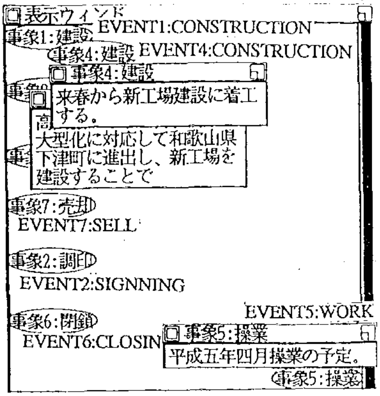

Figure 3: The classification result by the event occurrence time and the similarity

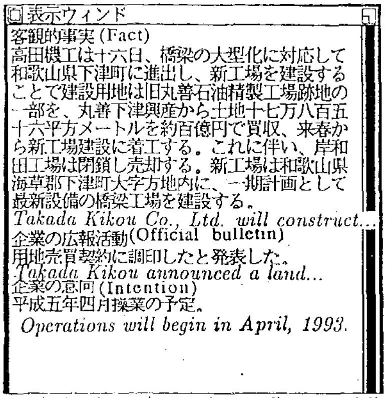

Figure 4: An abstraction result according to modality

3.2 An example of the abstraction

Figure 2 shows a typical abstraction result of ABEX. The events are classified by the event occurrence time and the simirality of each event. In this figure, x-axis indicates absolute event occurrence time and y-axis indicates relative similarity of events and circled icons indicate single events.

A typical classification result using the modality of information is shown in figure 3.

The Event 2 has the modality of an official bulletin and Event 5 has the modality of company intention, so we get the abstraction result shown in figure 3.

4 Conclusion

We have described a framework for a new abstraction method that utilizes classification. Classification is performed using the output of a semantic engine that is based on LIS form. Since the LIS form takes into account the incompleteness of knowledge, the system requires only a small amount of knowledge to perform the semantic analysis.

First, ABEX utilizes the selectivity of the semantic engine according to the domain and the event. Furthermore, ABEX classify according to the LIS constituents such as, TIME modality and so on. The generation mechanism is poor, but abstraction by classification is an easy way making an abstract. Furthermore, the classification method described here well supports human abstract tasks.

References

[1] M. E. Bratman. Plans and resource-bounded practical reasoning. *Computational Intelligence*, (4), 1988.

[2] C. J. Fillmore. Toward a modern theory of case. *Prentice-Hall*, 1969.

[3] U. Hahn. Topic essentials. *Coling-86*, 1986.

[4] H. Inagaki, S. Miyahara, and F. Obashi. Sentence disambiguation by document oriented preference sets. *Coling-90*, 1990.

[5] H. P. Luhn. The automatic creation of literature abstract. *IBM Journal, Vol2*, 1958.

[6] L. F. Rau. Information extraction and text summarization using linguistic knowledge acquisition. *Processing & Mannagment*, pages 419–428, 1989.